Temario

1. Introducción y mecánica

1.1 Presentación y objetivos
1.2 Comprobación del material
1.3 Torre Bot: pautas para construir el microbot
1.4 Mecánica: motores y estructura
1.5 Prueba del microbot con el programa CT293+

2. Programación I

2.1 Tarjetas CT6811
2.2 Programación en ensamblador del 68HC11: as11 y ejemplos sencillos
2.3 Programas de comunicaciones: puerto serie, norma RS-232, programas ctdialog, mcboot.
2.4 Tarjeta CT293+: etapa de potencia y sensores

3. Programación II y microbótica

3.1 Interrupciones : ejemplos y grabación en 68HC11-A1
3.2 Experimentos, concursos, otros microbots, otras tarjetas de control (uClinux).

4. Hardware de control

4.1 Drivers de la tarjeta CT293+. Puente en H
4.2 LDR's, potenciómetros y su programación
4.3 Motores paso a paso y drivers
4.4 Robots articulados: control en bucle cerrado
4.5 Programación cliente-servidor

5. Cierre

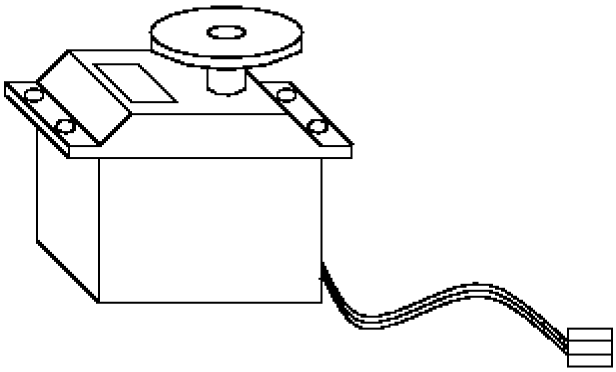
5.1 Utilización de los microbots en el futuro. Otros proyectos
5.2 Puesta a punto de los microbots y concurso final

La TorreBOT

Nivel de COOPERACIÓN	Comprende los sistemas en los cuales, a partir de un Nivel de Comunidad, se planifican o programan los microbots para que tengan conocimiento de la existencia de otros, tal que posean la capacidad de cooperar para el buen desarrollo de una tarea.
Nivel de COMUNIDAD	Se trata de la puesta en funcionamiento de más de un microbot dentro de un mismo entorno de forma simultánea y sin que ninguno de ellos tenga conocimientos explícitos de la existencia de otros en su mismo entorno.
Nivel de INTELIGENCIA	Abarca el planificador a largo plazo; en este nivel se introducen los objetivos del microbot que tienen relativa independencia de los sensores.
Nivel de CONTROL	Incluye los circuitos más básicos que relacionan las salidas de los sensores con las restantes unidades.
Nivel de REACCIÓN	Está formado por la batería de sensores independientemente de su naturaleza, así como los sistemas básicos para su manejo.
Nivel FÍSICO	Comprende la estructura física, las unidades motoras y las etapas de potencia.

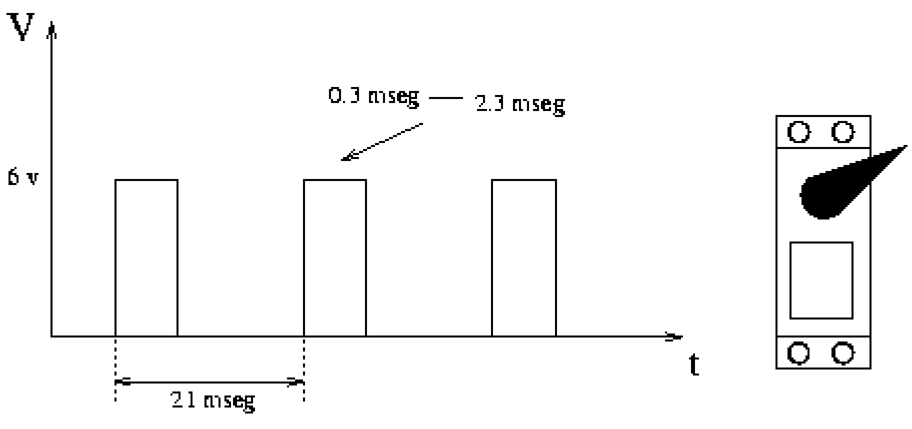
SERVOMECANISMO FUTABA S3003

- **Servomecanismo Futaba S3003**



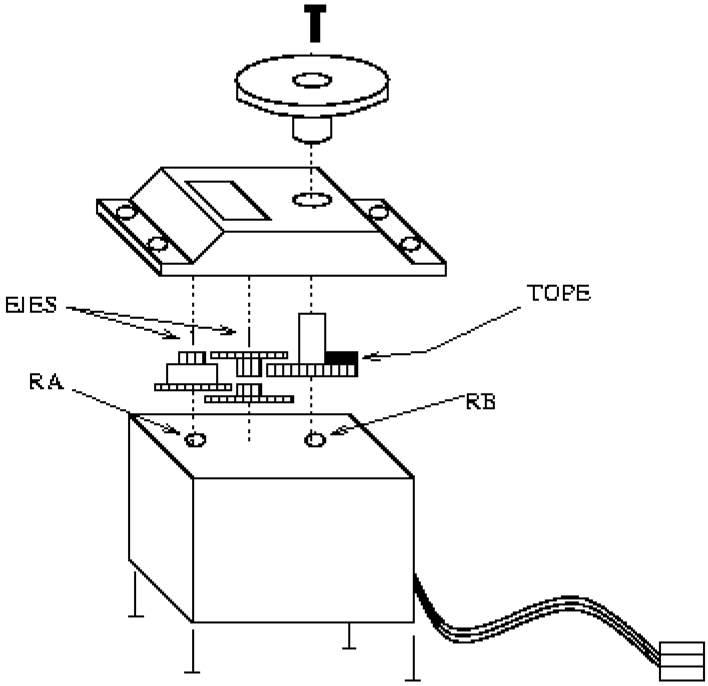
Motor utilizado en aplicaciones de aeromodelismo. Forma parte de una amplia familia siendo el más barato de todos. Nosotros lo utilizamos porque la relación calidad-prestaciones-precio es muy buena, compitiendo sobretodo con motores del tipo Maxon, Minimotor, etc

- **Señal de control del servomecanismo**

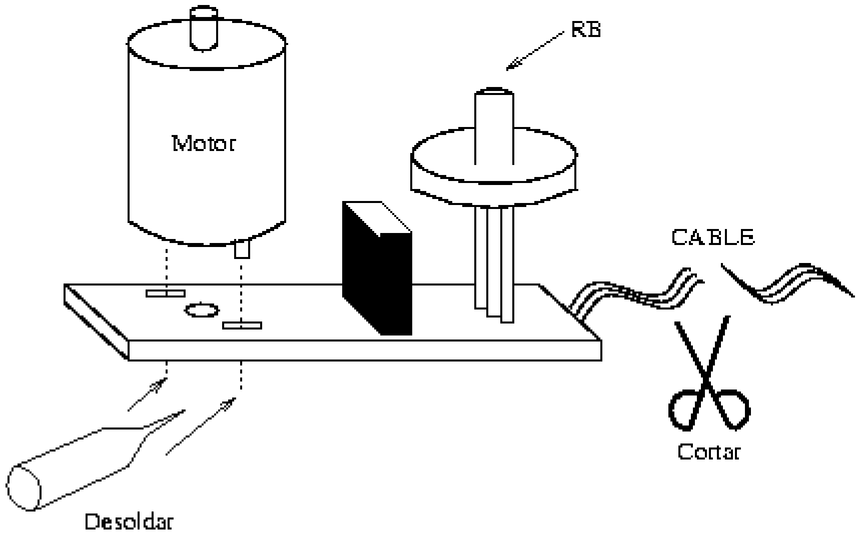


La señal que controla el servo es del tipo PWM. En este caso, el ancho del pulso en lugar de controlar la velocidad, lo más habitual en este tipo de señales, controla la posición. Los valores que definen el periodo y los anchos de la señal se representan en la figura de arriba.

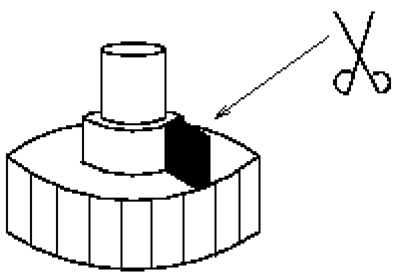
- **Caja de engranajes**



- **Desmontando el circuito de control**



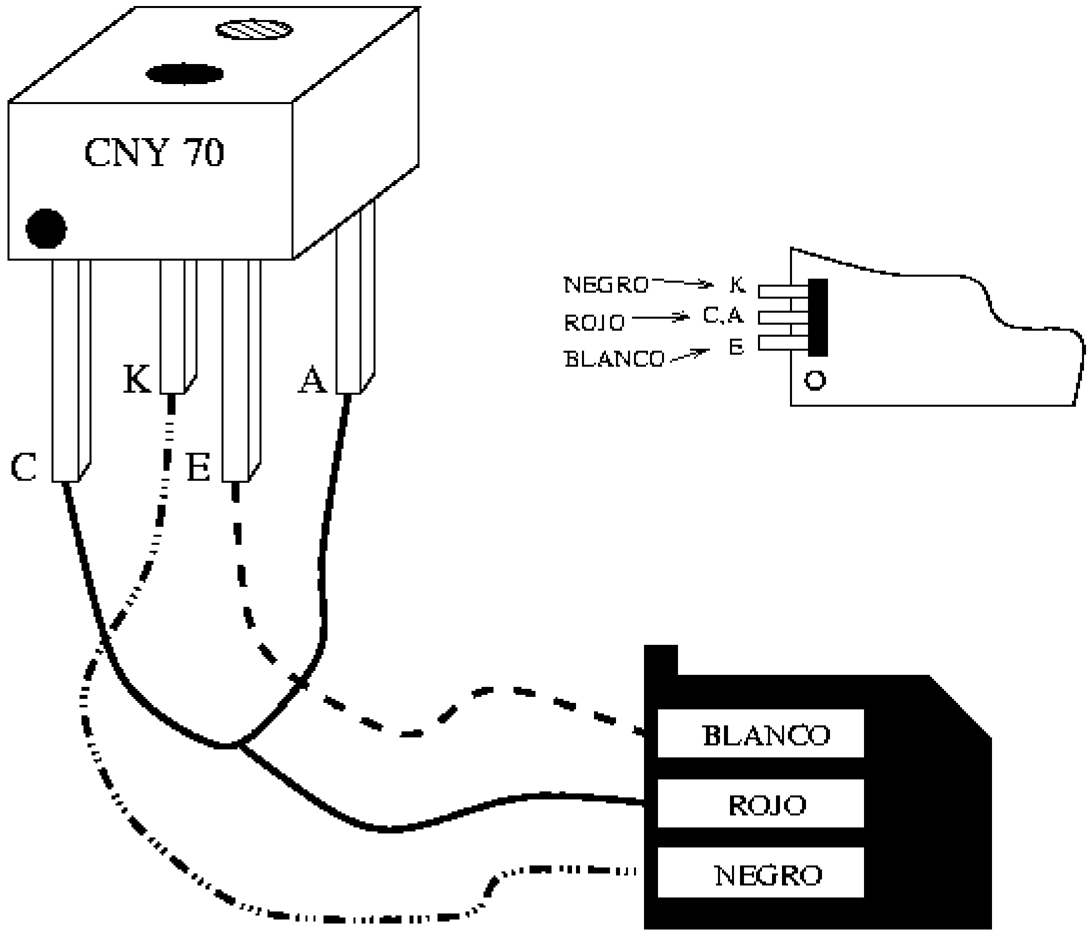
- **Eliminando el tope mecánico**



EJE DE SALIDA

LOS SENSORES DE INFRARROJOS

- Esquema y conexión a la CT293+



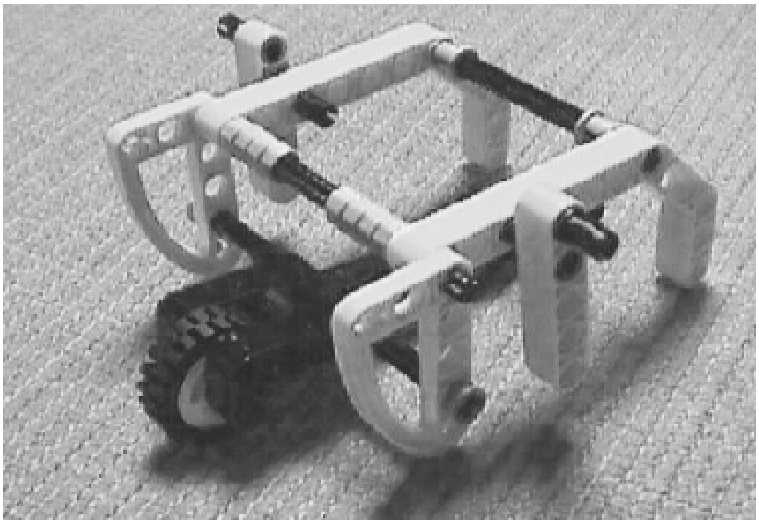
1. El sensor de infrarrojos es un CNY70. Sin entrar en detalles vamos a proceder a montarlo, en capítulos posteriores se explicará su funcionamiento y el circuito de polarización.

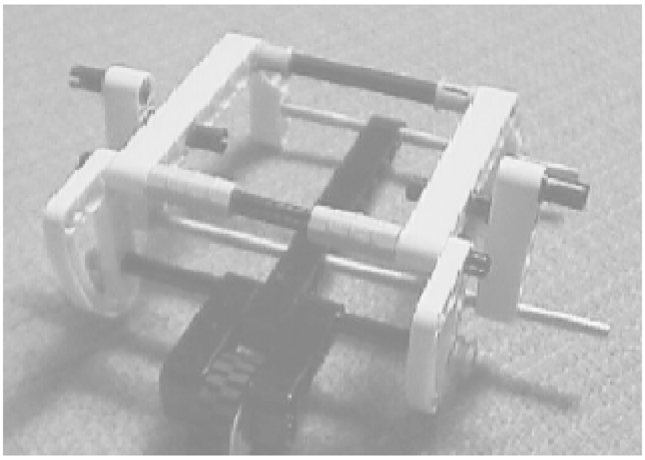
2. De las cuatro patas del sensor dos se conectan a masa por lo que las podremos corocircuitar, para ahorrarnos un cable de conexión.

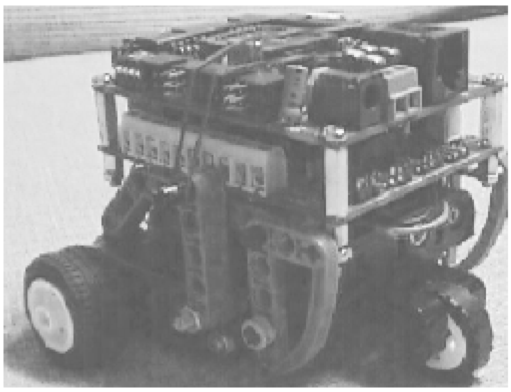
3. Utilizaremos el cable que ha sobrado del servomecanismo. Como tiene tres hilos soldaremos cada uno de ellos con cada una de las patas del sensor. Recomendamos seguir el esquema de la figura para que todos los sensores sean compatibles.

4. En el manual de usuario se recomienda conectar los motores y los sensores de una manera determinada para que todas las explicaciones que hay a continuación coincidan con la realidad. Si optamos por seguir el orden que nosotros queramos, tal vez la orden ir recto producirá que el robot gire a la derecha. :-))

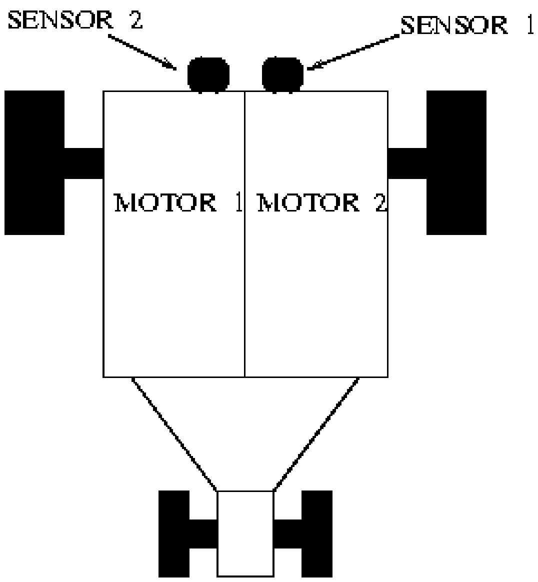
ESTRUCTURA DEL MICROBOT







CONEXIÓN DE LOS MOTORES Y SENSORES AL MICROBOT



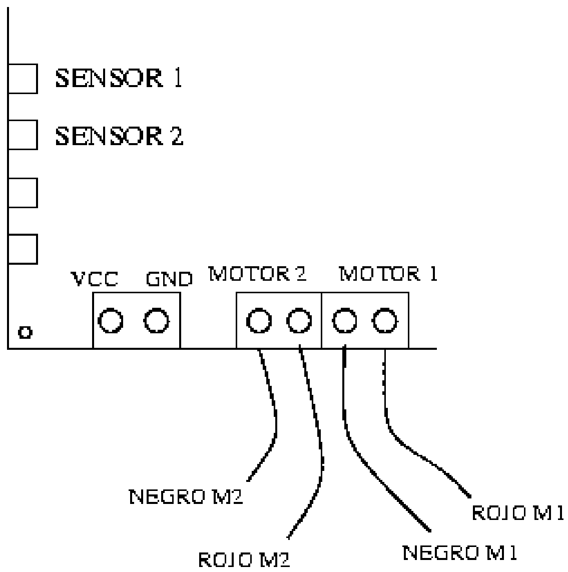
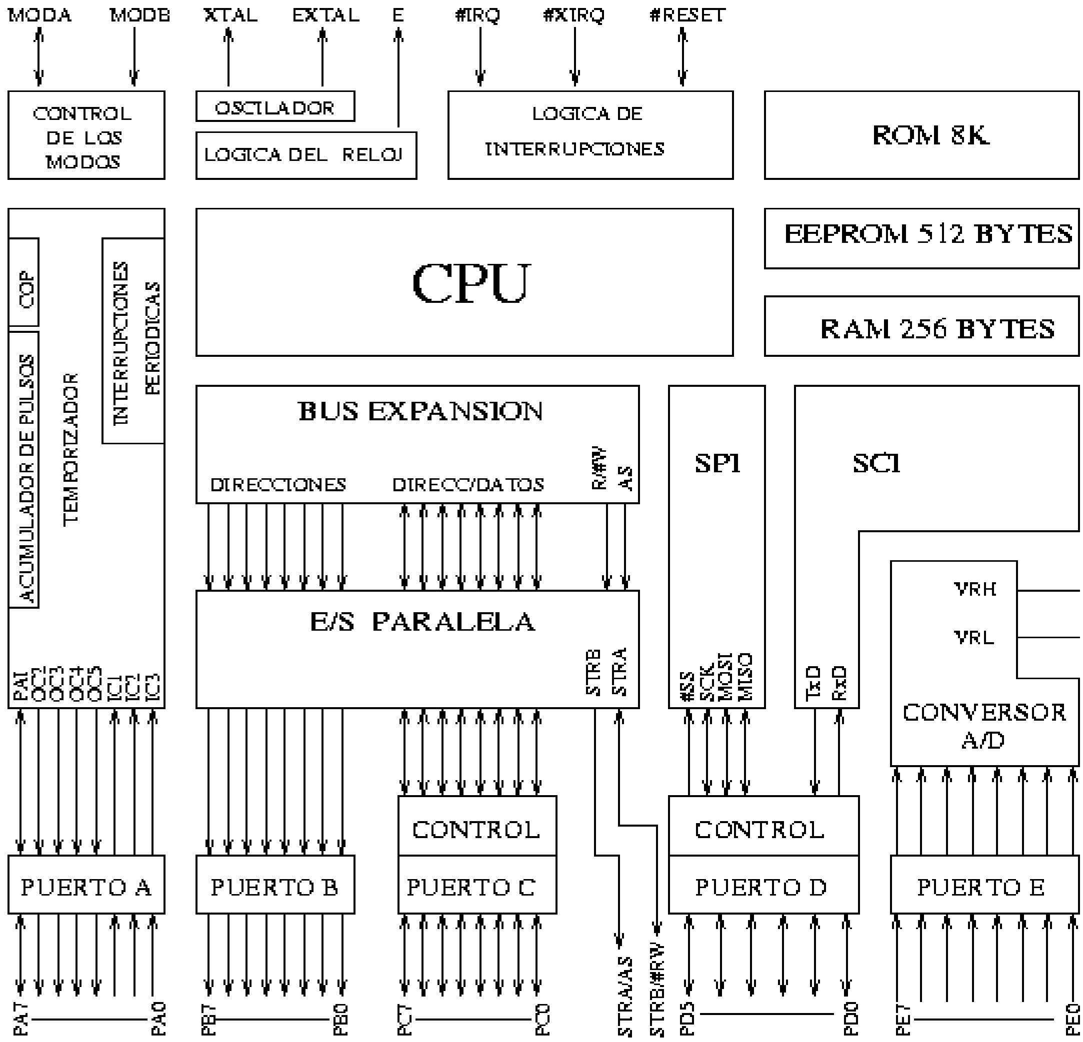
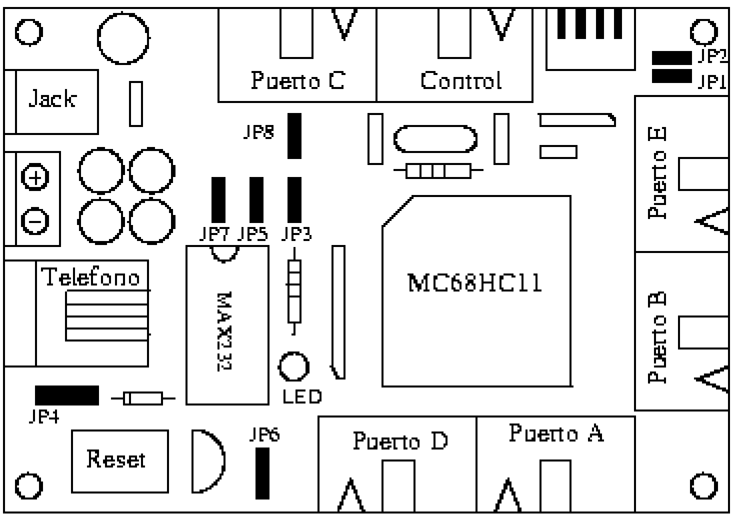


Diagrama de bloques del 68HC11 y de la tarjeta CT6811





Ensamblador del 68HC11 y comunicaciones serie

Programa LEDP:

```
; +------------------------------------------------------------------------+
; ¦ LEDP.     GRUPO J&J.                                       MICROBÓTICA ¦
; ¦------------------------------------------------------------------------¦
; ¦  Programa ejemplo para ser ejecutado en la tarjeta CT6811.             ¦
; ¦  Este programa se debe cargar en la RAM interna del 6811.              ¦
; ¦                                                                        ¦
; ¦    Simplemente se enciende y se apaga el led de la tarjeta CT6811.     ¦
; +------------------------------------------------------------------------+

        ORG $000

comienzo
        LDAA $1000
        EORA #$40         ; Cambiar de estado el bit PA6
        STAA $1000

        LDY #$FFFF        ; Realizar una pausa
dec     DEY
        CPY #0
        BNE dec

        BRA comienzo      ; Repetir el proceso

        END
```

Figura 26: Listado del programa LEDP.ASM

Compilador AS11 y programa de comunicaciones DOWNMCU:

```
C:\6811\CT6811>as11 ledon.asm
Freeware assember ASxx.EXE Ver 1.03.


Number of errors 0

C:\6811\CT6811>downmcu ledon -com2

DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.
Envío de programas a la entrenadora

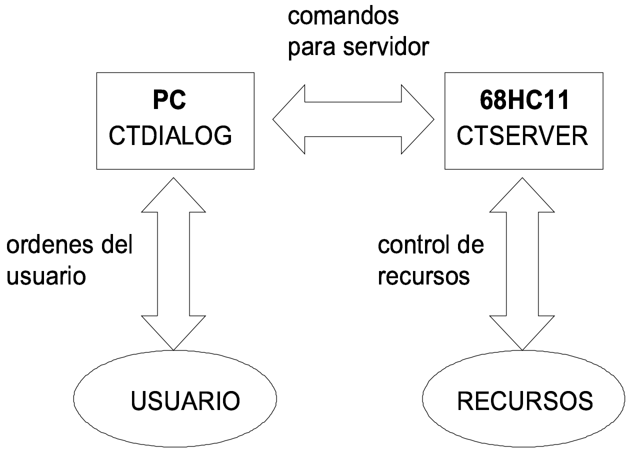
Fichero a enviar: .\ledon.S19
Puerto serie: COM2

Pulse reset en la entrenadora...
Transmitiendo: >>>>>>>.......................................................
.............................................................................
.............................................................................
..............................OK!
Envío correcto
Tamaño del programa: 7 bytes

C:\6811\CT6811>_
```

Figura 38: Ensamblado del programa LEDON.ASM y envío a la tarjeta CT6811

Programación de la tarjeta CT6811: programa CTDIALOG




```
C:\6811\CT6811>downmcu ctserver -com2

DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.
Envío de programas a la entrenadora

Fichero a enviar: .\ctserver.S19
Puerto serie: COM2

Pulse reset en la entrenadora...
Transmitiendo: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>......OK!
Envío correcto
Tamaño del programa: 250 bytes

C:\6811\CT6811>ctdialog -com2


CTDIALOG Versión 1.0 (C) GRUPO J&J. Diciembre 1996
Teclee HELP para obtener ayuda

Puerto actual: COM2
Estableciendo conexión con tarjeta..... conexión establecida

>
```

Figura 34: Ejecución del programa CTDIALOG.

```
>ms 1000 40
>md 1000 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
1000  40 00 03 00 00 FE 00 00 03 00 00 00 00 00 0D B9    @...............

1 bloque volcado

>ms 1000 0
>md 1000 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
1000  00 00 03 00 00 FE 00 00 03 00 00 00 00 00 18 38    ...............8

1 bloque volcado

>_
```

Figura 36: Encendido y apagado del LED desde el CTDIALOG

Programa de ejemplo: "menu.asm"

```
; +------------------------------------------------------------------------+
; ¦ MENÚ     (C) GRUPO J&J. Febrero 1997                                    ¦
; ¦------------------------------------------------------------------------¦
; ¦  Programa ejemplo para ser ejecutado en la tarjeta CT6811.             ¦
; ¦  Este programa se debe cargar en la RAM interna del 6811.              ¦
; ¦                                                                        ¦
; ¦  Ejemplo de cómo manejar un menú de opciones para programas interac-   ¦
; ¦ tivos con el usuario.                                                  ¦
; +------------------------------------------------------------------------+

CR      equ  13    ; Retorno de carro
LF      equ  10    ; Avance de linea

; Registros del SCI

BAUD    equ  $2B
SCCR1   equ  $2C
SCCR2   equ  $2D
SCSR    equ  $2E
SCDR    equ  $2F

        LDX #$1000       ; Para acceder a registros del SCI

bucle   LDY #menu
        BSR send_cad     ; Sacar menú
wait    BSR leer_car     ; Leer tecla
        CMPA #'1'
        BEQ opcion1      ; Tecla '1'--> Opción 1 del menú.
        CMPA #'2'
        BEQ opcion2      ; Tecla '2'--> Opción 2 del menú.
        BRA wait

opcion1 LDAA $1000
        EORA #$40        ; Cambiar estado bit 6 puerto A
        STAA $1000
        BRA wait

opcion2 BRA bucle        ; Volver a sacar el menú


;+-----------------------------------------------------+
;¦ Rutina par leer un carácter del puerto serie (SCI)  ¦
;¦ La rutina espera hasta que llegue algún carácter    ¦
;¦ ENTRADAS: Ninguna.                                  ¦
;¦ SALIDAS: El acumulador A contiene el carácter recibido ¦
;+-----------------------------------------------------+
leer_car BRCLR SCSR,X $10 leer_car   ; Esperar hasta que llegue un carácter
        LDAA SCDR,X
        RTS
```

```
;+--------------------------------------------------------+
;¦ Enviar un carácter por el puerto serie (SCI)            ¦
;¦ ENTRADAS: El acumulador A contiene el carácter a enviar ¦
;¦ SALIDAS: Ninguna.                                      ¦
;+--------------------------------------------------------+
enviar  BRCLR SCSR,X $80 enviar
        STAA SCDR,X
        RTS

;+--------------------------------------------------------+
;¦ Enviar una cadena de caracteres por el puerto serie.   ¦
;¦ La cadena debe terminar con el carácter 0              ¦
;¦ ENTRADAS: Registro Y contiene dirección cadena a enviar ¦
;¦ SALIDAS: El acumulador A contiene el carácter recibido  ¦
;+--------------------------------------------------------+
send_cad LDAA 0,Y        ; Meter en A el carácter a enviar
         CMPA #0         ; ¿Fin de la cadena?
         BEQ fin         ; Si--> retornar
         BSR enviar      ; NO--> enviar carácter.
         INY             ; Apuntar a la sig. posición de memoria
         BRA send_cad    ; Repetir todo
fin      RTS


;+--------+
;¦ DATOS  ¦
;+--------+
menú    FCB CR,LF,LF
        FCC "       MENÚ DE OPCIONES"
        FCB CR,LF
        FCC "       ================"
        FCB CR,LF,LF
        FCC "  1.- Cambiar de estado el LED"
        FCB CR,LF
        FCC "  2.- Sacar este menú"
        FCB CR,LF
        FCC "  Opcion: "
        FCB 0

        END
```

Puerto A: Control Motores y sensores

1. El puerto A se localiza en la dirección **$1000** del mapa de memoria.
2. El significado de sus bits es el siguiente:

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
SENSOR 3 SWITCH 2	MOTOR 2 DIR	MOTOR 1 DIR	MOTOR 2 ON/OFF	MOTOR 1 ON/OFF	SENSOR 4 SWITCH 1	SENSOR 2 SWITCH 3	SENSOR 1 SWITCH 4

3. La interpretación del bit de un sensor es la siguiente:

   bit a **NIVEL ALTO '1'** se ha detectado **NEGRO**
   bit a **NIVEL BAJO '0'** se ha detectado **BLANCO (No negro)**

4. La interpretación de los bits de los motores es:

	MOTOR 1	***MOTOR 2***
DIRECCIÓN	BIT 5: ON (1) -> Derecha OFF (0) -> Izquierda	BIT 6: ON (1) -> Izquierda OFF (0) -> Derecha
ESTADO	BIT 3: ON (1) -> Motor ON OFF (0) -> Motor OFF	BIT 4: ON (1) -> Motor ON OFF (0) -> Motor OFF

5. Si se han conectado los motores y sensores como se dijo al principio:

Sensor 1 PA0	***Sensor 2 PA1***	***Acción Microbot***	***Estado Motor 1***	***Dirección Motor 1***	***Estado Motor 2***	***Dirección Motor 2***	***Byte***
1	1	Avanzar	1	0	1	0	$18
0	1	Izquierda	1	1	1	0	$38
1	0	Derecha	1	0	1	1	$58

6. Ejemplo: Programa linea recta.

Puerto E: Entradas analógicas

1. El puerto E se localiza en la dirección $100A del mapa de memoria.
2. El significado de sus bits es el siguiente:

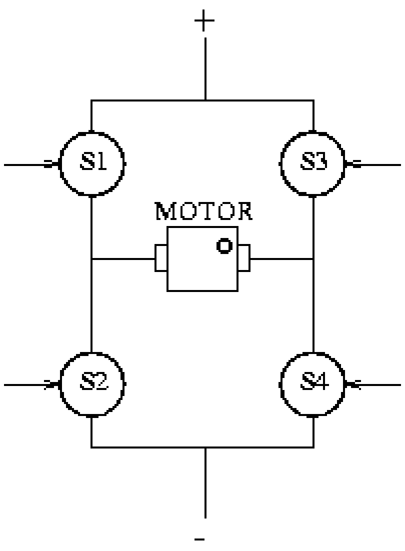
Otros sistemas/experiencias

- Experimentos. Velas
- Concursos
- Sistemas basados en microcontrolador
  - Control de trenes
  - Cerraduras electrónicas



HARDWARE DE LA CT293+

• DRIVER DE POTENCIA PARA LOS MOTORES

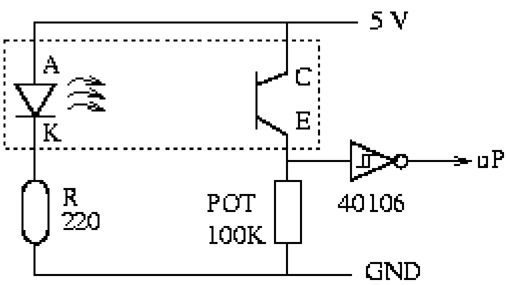


No se puede conectar el motor directamente a la tensión positiva y negativa porque es necesario que éstas se puedan intercambiar para hacer girar al motor en ambos sentidos. Para solucionar lo anterior se utiliza un circuito conocido como **puente en H**.

S1	***S3***	***S2***	***S4***	***Acción Motor***
ON	OFF	OFF	ON	Derecha
OFF	ON	ON	OFF	Izquierda
ON	ON	OFF	OFF	Parado
OFF	OFF	ON	ON	Parado
OFF	OFF	OFF	OFF	Parado
ON	ON	ON	ON	Error

• POLARIZACIÓN DE LOS SENSORES DE INFRARROJO

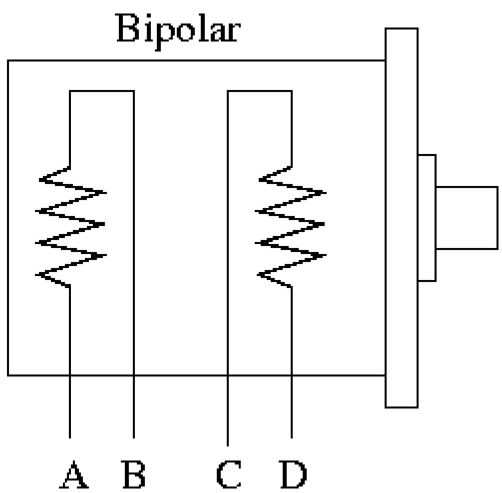
Los CNY70 se pueden polarizar de dos formas: En continua. (Utilizada en la CT293+) o con modulación de pulsos. Mediante la primera el circuito es muy sencillo y se obtienen muy buenos resultados a la hora de distinguir entre negro y blanco. La segunda opción permite un mayor alcance pero se complica el circuito y el software.

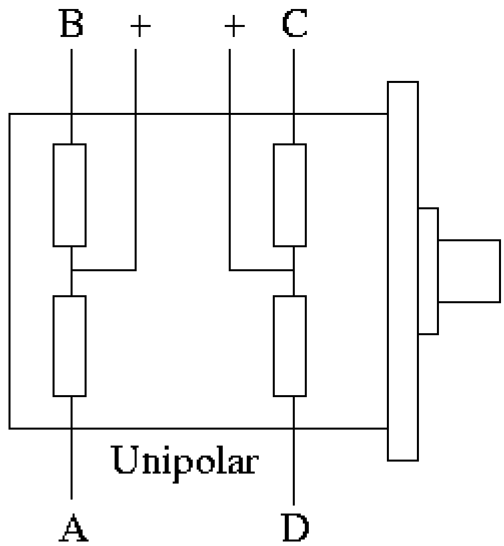


MOTORES PASO A PASO

En un motor de CC el eje se puede detener en cualquier ángulo de giro, mientras que en uno PP sólo puede situarse en unos determinados. En concreto **' = + n. '** , donde 'n' corresponde al número de pasos dados desde una posición de partida. Generalmente los motores PP se caracterizan por el número de pasos que hay que dar para girar 360º, por ejemplo los hay de 48 y 200 pasos.

Lo bueno de estos motores es que para saber la posición no hace falta cerrar el bucle, tan sólo conocer la posición de partida. Lo malo es que consumen más que uno equivalente en CC, y que no giran tan rápido como los de continua.





Para mover el eje de estos motores hay que ir avanzando una posición de la tabla siguiente. Según se haga en orden ascendente o descendente girará a la derecha o a la izquierda.

BIPOLAR	*A*	*B*	*C*	*D*
1	-	+	-	+
2	+	-	-	+
3	+	-	+	-
4	-	+	+	-

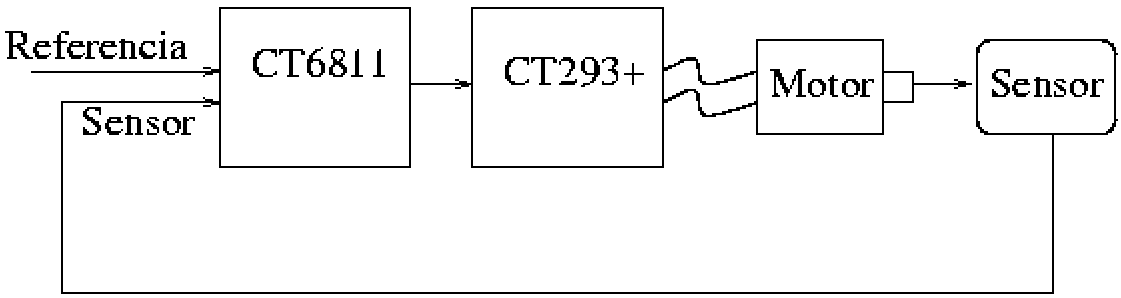
UINPOLAR	*A*	*B*	*C*	*D*
1	-		-	
2		-	-	
3		-		-
4	-			-

CONTROL EN BUCLE CERRADO

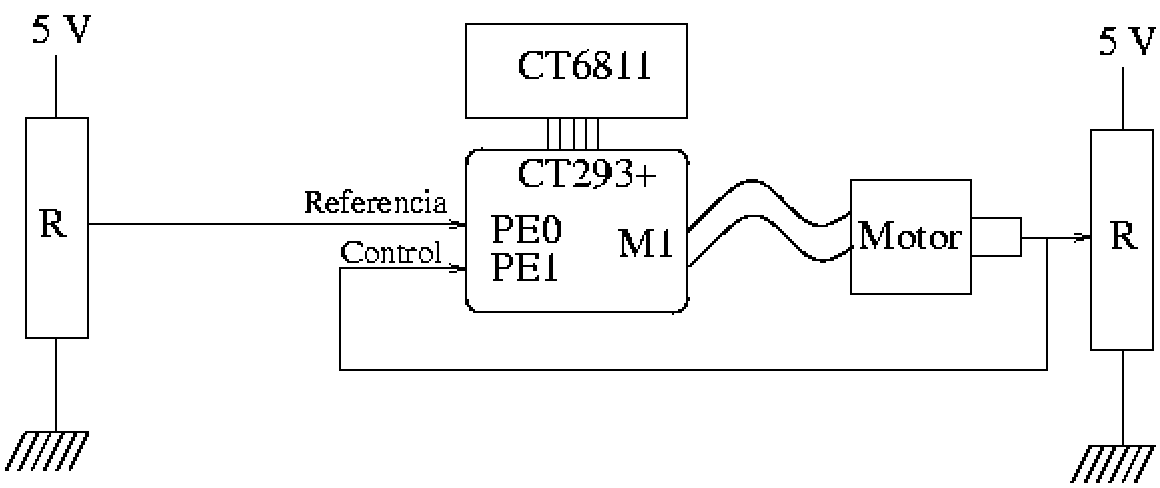
El control en **bucle cerrado** es necesario para la mayoría de las aplicaciones de robótica. Este tipo de control nos va a permitir conocer el estado de los motores y actuar sobre ellos cuando se perciba alguna variación no deseada de éste.

Con los motores paso a paso se tiene un control en **bucle abierto**, fácil de implementar pero mantiene un alto grado de incertidumbre. Cualquier modificación por error del eje del motor no será contemplada en el algoritmo de control y eso traerá errores.

La técnica descrita aquí se puede emplear tanto en motores *paso a paso* como en los de *continua*, pero se suele aplicar sobre éstos últimos.



- **Ejemplo práctico: Control de posición con potenciómetro.**



Se va a realizar un ejercicio práctico, mediante un potenciómetro controlaremos la situación del eje de un motor. El potenciómetro se encargará de generar la señal de referencia, que más adelante podrá ser sustituida por una generada por ordenador o directamente por la CT6811.